NOνA Database Requirements

Jonathan M. Paley, Julie Trumbo, Gerald Guglielmo,
Igor Mandrichenko, Andrew Norman

January 29, 2007

1 Introduction

During the early stages of the NOνA project, measured properties of the various components of the detectors, as well as progress of the construction of each detector will be tracked via a database and a web interface. At later stages of the project, the performance of the NOνA detectors will also be tracked in the database for both operations and offline analysis of the data. This document will outline the hardware and software requirements for the NOνA database.

2 Database Hardware Requirements

The requirements for the NOνA database hardware are as follows:

1. Support for hardware must be 24/7, minimizing downtime.
2. Vendor support for 24/7 hardware maintenance will be established.
3. Hardware failover should be implemented, by providing a backup system available for use.
4. A Linux operating system, certified with the database will be implemented.
5. Detector database hardware will be positioned in MN and at Fermilab.

3 Database Software Requirements

3.1 Generic Requirements

1. The database for Nova must be a robust, reliable engine capable of operating at 99%+ scheduled uptime.

2. After data has been written to the database, (a commit), there should be no data loss.

3. A cache system must be available to store data during times the hardware or database is unavailable.

4. Must sustain 10 simultaneous online connections.

5. Data will need to be available for 20 years.

6. Operating system calls from the database are not required, however, store procedures, triggers, and binary data types are required.

7. The database must scale to approximately 1 Terabyte.

3.2 Security

1. Doe and Fermilab mandated security baselines will be followed for both the operating system, hardware and database.

2. Regularly scheduled downtimes will be allowed for patching. Minimally these will be available quarterly in sync with the security patch releases for the operating system and database.

3. Non quarterly downtimes will be granted for emergency patching as required. These are normally very minimal.

4. At lease one user account should be created that will have global read (search) permissions only (eg, username: "novadbread").

3.3 Support

1. 24/7 tech support

2. Security patches

3. Prioritization for production issues

4. Assistance with proven backup recovery techniques

5. Timely bug reporting and resolution

3.4 Table Requirements

1. User permissions for table create, insert, update and delete will be given upon request.

2. Every table will have the following columns:

   (a) create_date (timestamp)

   (b) create_user

   (c) update date (timestamp)

   (d) update_user

3.5 Database Interface Requirements

As components of the NOνA detectors are manufactured, tested, delivered and possibly retested, physicists and technicians responsible for each component will need to insert data into the NOνA hardware database. Currently we expect to allow users to insert, update, delete, search and view the data via a web interface. The requirements for this interface are:

1. valid users only may view information in the database

2. users should be able to view and sort data in tables (see Sec. 3.5.1 for details), and have the ability to edit this data (if the user has the proper permissions)

3. users should be able to upload data into tables from comma separated value (CSV) files (eg, generated by spreadsheet)

4. users should be able to download data from tables into a CSV file

5. users should be able to plot data from tables (see Sec. 3.5.2 for details)

6. users should be able to generate reports from tables (see Sec. 3.5.3)

Note that 1-4 are required for the first release of the NOνA Hardware database interface.

3.5.1 Viewing Table Data

We expect that users will want to view data in tables via the web interface. We also expect that users will wish to view only a subset of the data. Therefore, the user should be able to both select the columns they wish displayed as well as limit the amount of data shown according to:

1. create[update]_date
2. date_shipped
3. date_received
4. create[update]_user

Furthermore, when displaying data, the user should be able to select the column on which to sort. By default, the sort should be done in descending order, such that the most recent dates are shown first. Finally, we expect that users will wish to edit data in the database tables. Using the above feature of viewing data, users with proper permissions should be able to "find" the proper row to edit, make changes and update the table.

3.5.2 Plotting Data from Tables

We expect that users will to see or create plots of data in the tables. There will be a number of standard plots to be used as "metrics", however users may also want to create plots that are not standard. Therefore we will want to allow users to select two or three columns to plot against each other. Furthermore, users should also be able to limit the ranges of the data to be plotted.

Each table will likely have it's own set of standard metric plots, however the following should be considered standard for all tables:

1. num. entries vs. date_shipped
2. num. entries vs. date_received
3. distribution of (date_received-date_shipped)
4. (date_received-date_shipped) vs. date_received

There should be links for users to click on that will automatically generate these plots.

Users may wish to select two (or three?) different columns to plot against each other. In this case, users should only be allowed to select columns that make sense to plot, for example integers, floats, and time stamps. User name may also be a possibility, however this is not a requirement.

All plots should:

1. be a .gif or .png data format,
2. have a clear title,
3. have axes labeled clearly; font size should be large and spacing between tick marks should be large enough to accomodate clear labels,

Users should also be able to plot different data sets on the same graph. As an example, users may want to plot different attenuation length measurement data sets against each other, or plot num. entries vs. date_shipped and num. entries vs. date_received on the same graph for easy comparison. Under these circumstances, each data set should be represented by different line colors, line type and/or markers.

3.5.3 Generating Reports

The basic idea here is the same as in Sec. 3.5.2, however instead of generating a graph, the data will be dumped to a text file.